SYSTEM FOR TRANSFERRING PERSONALIZE MATTER FROM ONE COMPUTER TO ANOTHER

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit under 35 U.S.C. § 119(e) of U.S. Provisional Patent Application No. 60/156,638, filed September 29, 1999, and U.S. Provisional Patent Application No. 60/214,504, filed on June 28, 2000, and claims the benefit under 35 U.S.C. § 120 of U.S. Patent Application No. 09/676,328, filed September 29, 2000, all of which applications are hereby incorporated herein by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] This invention has been created without the sponsorship or funding of any federally sponsored research or development program.

BACKGROUND OF THE INVENTION

[0003]

FIELD OF THE INVENTION

[0004] This invention relates to computer voice recognition enhancements. It explains methodologies for measuring reliability, accuracy, and performance as system responsiveness using a standardized method of measurement. The invention introduces a method of machine independent user mobility between different voice recognition systems. It addresses a method for enabling speaker independent voice recognition for masses of people without the need for training or enrollment. It describes how to apply the technology to a new style of interactive real time voice to text handheld transcriber including visual feedback to replace the previous handheld transcribers that are only recording devices. It describes using these techniques in a system that translates voice mail audio into text readable messages.

FIELD OF THE RELATED ART

[0005] Prior to voice recognition dictation, transcription was completely done by humans. Some inventions enhanced this situation by providing a means to make it faster and easier for the transcriptionist to work with the audio file that needed to be transcribed. An example is US patent number 6,175,822 Bryce Alan Jones (Method and System for Providing Network Base Transcription Services) where the audio file is captured at one location and sent over the Internet and played back to a transcriptionist at a second location removing the requirement of the transcriptionist to be at the location where the dictation was taking place. Over time features were added to audio files including a parallel processing using speech recognition. An example of this is US patent number 6,073,103 Dunn et al. (Display Accessory for Record Playback System) where it is described how to combine audio voice input and speech recognition applications to identifying numbers in the voice audio files. This gives a user the ability to index into the audio where the numbers are located. Another feature added was the ability to capture audio when the speech recognition was turned off to avoid loss of any of the spoken words, as described in US patent number 6,415,258 Reynar et al. (Background Audio Recovery System).

[0006] In general terms however, voice recognition dictation products that are presently in the market follow the typical clone PC market strategy. The state of the art is buying a personal computer that is designed as a general purpose computing device, installing voice recognition software (i.e. IBM ViaVoice, L&H Voice Express, Philips Speech Pro from Philips, Dragon Naturally Speaking, from Dragon Systems), and using that configuration as a Large Vocabulary Voice Recognition dictation system. When using Large Vocabulary Voice Recognition (LVVR) applications in the clone PC environment, two problems are experienced: machine dependency and speaker dependency. While this approach is typically used throughout the computer industry, it often leaves users frustrated with accuracy and performance of the voice recognition applications.

[0007] This is especially true when applying the technology to a hand held transcriber type of devices like a tape recorder or digital voice recorder. The industry standard for handheld dictation is to use handheld tape recorders or memory devices that provide the same functionality as tape recorders, i.e. a handheld digital reorder. Voice recognition software packages supported connections from these handheld devices to desktop types of computers allowing the voice to be translated into text through a voice recognition package like IBM's ViaVoice voice recognition software. These approaches have many problems including: No direct feedback while the dictation is taking place, it was not real time large vocabulary voice recognition, training for the voice recognition was cumbersome to accomplish resulting in poor accuracy and user frustration, and training required redundant work since a separate voice model is needed from the desktop speaker voice files. Moreover, updating the voice parameters and training was typically not possible or very difficult to accomplish resulting in the accuracy level not getting better over time. And lastly, a separate physical connection to the dictation device was needed to accomplish the translation to text with little to no control of manipulating the text output until the entire recorded voice was dumped and translated into text.

[0008] The voice recognition dictation systems require the training sessions to enable the system to identify the words of a person is speaking. The process of training a voice recognition system creates speaker voice files or a "Voice Model". A "Voice Model" is defined here as a signal, information, or electronic data file that is information and/or parameters that representation of a person's voice or a noise. A Voice Model contains attributes that characterize specific speaking items such as formants, phonemes, speaking rate, pause length, acoustic models, unique vocabulary's, etc. for a given user. One use for a voice model that contains data and parameters of a specific user is that it allows the user to take advantage of Large Vocabulary Voice Recognition (LVVR) dictation applications. All approaches to LVVR (e.g. Acoustic phonetic, Pattern recognition, Artificial intelligence, Neural networks, etc.) require some training. Training is required to create a reference pattern from which decisions are made using templates or statistical models (e.g. Markov Models and Hidden Markov Models) as to the probability of the audio word to be translated to a text displayed word. When using Large Vocabulary Voice Recognition applications, training of the voice recognition software allows the software to identify words during the uniqueness of a specific person speaking. Since training can be time consuming and ongoing task and typically results in speaker dependency other inventions have avoided confronting the training and speaker voice models issues needed to accomplish speaker independent and/or mobility between voice recognition dictation systems. As an example, the problem exists and was described within US patent number 5,822,727 Garberg et al. (Method for Automatic Speech Recognition and Telephony) where voice recognition training is accomplished using sub-words of a current speaker compared with templates for a plurality of speakers. This patent recognizes that there is a need for a more convenient and thorough process for building a database of sub word transcriptions and a database using speaker independent templates.

[0009] US patent number 6,477,491 Chandler et al. describes needing training for voice recognition applications but does not provide any specific means to accomplish this task and is focused on providing identity of a specific person by the specific microphone they are speaking into.

[0010] Therefore it is generally accepted that upfront training to gain an acceptable level of accuracy and system responsiveness requires time and effort as the system learns a specific user. This investment of time and effort is a per machine cost adding to machine dependency. Training a voice recognition system will result in a specific system voice to text translation accuracy in a given time indicating system responsiveness/performance. When trying to determine and obtain the highest level of system accuracy and performance, one can spend much effort, time, and money trying to determine the best options, (performance and accuracy versus components, effort, and cost). This has led to frustration and funds wasted with the result being that the speech recognition system is left sitting on the shelf or discarded.

[0011] Many professional people use more than 1 computer to accomplish their daily task. When more than 1 computer is used for voice recognition, accuracy and performance may not be consistent due to different levels of training accomplished for each system. This was discovered through experimentation with voice recognition packages and was verified in talking with doctors, lawyers, and other professionals that use speech recognition. These users described accuracy for example, at an estimated 94 percent but all claimed that they didn't know accurately what the accuracy was. Other statements made included how accuracy would vary when using an assortment of machines for voice dictation.

[0012] This invention is targeting to resolve the specific problems of measuring a standard performance and standard accuracy, machine dependency, speaker dependency, mobility, and methods of estimating accurate cost for users and manufacturers.

BRIEF SUMMARY OF THE INVENTION

[0013] This invention includes several components that provide enhancements and ease of use features to voice recognition systems and applications. In accordance with this invention, it is possible to reliably measure accuracy and responsiveness of a voice recognition system used for dictation purposes. With the ability to measure these key metrics other enhancements can then be added to voice recognition systems with a quick and easy determination of system improvement or degradation. One such enhancement described is the ability to move speaker voice models (Voice Modeled Mobility) between systems with the ability to quickly determine the success of a quick user enrollment versus a full training session of a voice recognition system. The measurements can also be applied to a new type of handheld transcriber with internal voice dictation software eliminating the need for a two-step process of recording the dictation notes and then transferring them to the voice recognition software for a speech to text translation. Further advantages can be achieved by applying the RAP Rate measurement techniques to engineering and manufacturing processes resulting in products that have a common reference and relationships providing a known value to industry, users, and customers prior to purchasing the voice dictation product. Applying the RAP Rate measurement techniques with other techniques for determining voice recognition user speech patterns (described in detail later) enables the creation of a new type of speaker voice model or a (Super Voice Model) that can be applied to many people without the prerequisite of training or voice recognition system enrollment. In overview this invention includes components that measure voice recognition metrics (RAP meter), provide ease of use for the movement of speaker voice models (Voice Model Mobility), a handheld transcriber that includes voice recognition software for dictation internal to the transcriber (Powerful Handheld Device), a process for the manufacturing and verification of systems used for voice dictation purposes (RAP Rate Manufacturing Process), a methodology for creating speaker independent voice models (Super Voice Model), and applying these techniques of RAP Meters, Voice Model Mobility, Super Voice Model, and Powerful Handheld Devices to a Audio Voice Mail to Text Translation system. These components and their related features and advantages are described in the description of the preferred embodiment using the drawings as reference.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is the opening screen of a software application called Voice Model Mobility. It displays the voice recognition application being used (current speech engine) and its related software version. It also has four buttons for controlling the application including moving speaker voice models from the voice recognition application (Move voice model from speech engine), moving voice models to the speech recognition application (Move voice model from media), and help and exit buttons.

[0015] FIG. 2 illustrates the user control screen of the Voice Model Mobility software used when copying a voice model from the speech engine to media (disk drive, tape drive, writable CD, network, or other transfer medium). It has for user control's including voice model selection button, destination button, and OK and cancel button.

[0016] FIG. 3 displays the user control screen of the Voice Model Mobility software used when moving a voice model from media into a speech recognition application. This control allows the location of the voice model to be selected and OK and cancel button.

[0017] FIG. 4 displays the user control screen of the Voice Model Mobility software used when moving a voice model from media into a speech recognition application. There is a button to select the voice model and OK and cancel buttons.

[0018] FIG. 5 indicates a Voice Model Mobility error handling dialog box example indicating to the user attempting to move voice model that the voice model was not successfully moved.

[0019] FIG. 6 illustrates the RAP Meter software opening screen. It contained six buttons and a display area to provide visual feedback to the user. The "Verify RAP Rate" button launches a user screen to perform the RAP Rate test. "The Select Mic" button allows the tests user to select a specific microphone on the system being used. The "Log File" button is a file that the user can review to see specific details of the Rap Rate test. The "Certificate Button" displays a certificate that can be shared with others indicating what level of Rap Rate the system under test achieved. The "Help" button displays the Rap Meter online help documentation, and the "Exit" button exits the RAP Meter program.

[0020] FIG. 7 illustrates the first opening screen of a RAP Rate test session. This screen enables the user to input specific information about the test to be performed including test name, voice recognition software used, and specific hardware options like the microphone or sound input port to be used for the test.

[0021] FIG. 8 is the RAP Rate user interface where the testing is implemented. It contains three display areas and two "New Test" buttons that start either the accuracy or the performance test. The "Delay" display illustrates the response time of how long it takes the speech recognition software to translate an audio spoken word into text display on a computer screen. The "Text" of display area provides the text to be read during the RAP Rate testing. The "Performance" display area provides the text to be read for the performance test. The "Log File" displays a log file of the current test. The "OK" button reverts back to the RAP Meter main screen, and "Cancel Button" reverts back to the main screen.

[0022] FIG. 9 is a display of the RAP Rate Certificate. The RAP Rate certificate is provided by the RAP Meter after the RAP Rate test has been completed, used for sharing and displaying the RAP Rate achievement for a specific system.

[0023] FIG. 10 is an example of a RAP Rate log file From a Typical RAP Rate test run. It illustrates the detailed test results for the performance in the accuracy test and also includes system specific configuration.

[0024] FIG. 11 shows a microphone and a hand held transcriber connected to a voice recognition system using a Y connector cable enabling simultaneous training of the voice recognition system for a microphone and transcriber input devices.

[0025] FIG. 12 is a prototype handheld computer with voice recognition software included for the purposes of voice dictation.

[0026] FIG. 13 illustrates computer hardware and the relationship of the components. The components that are shown fit into the form factor of a handheld transcriber.

[0027] FIG.'S 14 and 15 illustrate a flow chat of a manufacturing process using the RAP Rate metrics to produce voice recognition systems with an observed and predictable level of accuracy and responsiveness.

[0028] FIG. 16 illustrates a process sheet to support the RAP Rate manufacturing process.

[0029] FIG. 17 illustrates the major components and overview process for a Super Voice Model (SVM).

DETAILED DESCRIPTION OF THE INVENTION

[0030] It is the object of this invention to provide a method for transferring voice models defined as Voice Model Mobility. Voice Model Mobility (VMM) was originally conceived due to the problem of having to train multiple voice recognition dictation machines for a single person's voice. This was discovered when experimenting with voice recognition dictation applications. It was determined that a better way to use multiple machines was to separate the files and parameters that characterize the user, package the files and parameters as a voice model and move them to a medium for transfer and installation into another separate system. Voice models and a means to package, move them, and install them can and should be independent of the voice recognition applications allowing the owner of a voice model the ability to plug into and use any voice recognition machine. Voice models and training are assumed needed and can be time-consuming therefore; voice recognition applications provided backup mechanisms to restore voice files to their original locations. They did not however, provide a means to transfer voice models between systems. Prior to VMM, moving voice models between voice recognition systems did not exist and there was no easy way to move these specific user parameters and data between machines. Several experiments were done in effort of understanding why the voice recognition applications did not support such features. From these experiments it was discovered that the lack of ability to create and move a voice model was not technical. The first experiment was to use the backup and save feature provided with the Dragon Professional voice recognition application. The problems encountered when trying to accomplish this included a different filename when restoring the user from when the user was saved. Another problem was the limitation of where the backup could be saved. In other words the voice model was not mobile. The second experiment was to copy the voice model files directly to another location and then copy them back to use them. In some cases this approach appeared to work although it took some trial and error until the exact files that needed to be copied were discovered. Crashes and hangs occurred often. Problems encountered prior to successful file copies included; user voice files contamination, the system hanging when trying to open a specific user, or the Dragon application no longer finding the user for opening. Although this approach sometimes yielded success it was discovered that the user would have to be created first, and then the files could be copied. This was due to registry entries not being setup as part of the copy process. A Visual Basic prototype was coded using this method for user interface experimentation. The third effort included investigation of the system registry to determine if Dragon was setting any parameters using the registry. This was found to be true and solved the final problems. The current version of VMM is coded in the C programming language.

[0031] The ability to unplug these parameters and data from one machine and transfer the files to another machine using mediums like disk, optical disk, floppy, network, etc. is defined here as Voice Model Mobility (VMM). Voice Model Mobility as described here is a software application. However, people skilled in the art will recognize that the specific functionality could be included as part of a hardware pluggable device. Referencing Figures 1 through 5, Figure 1 illustrates the start up screen from voice model mobility software. The opening screen contains two buttons to control the movement of voice models between media and voice recognition systems. Figure 2 illustrates another screen of the user interface. This screen is displayed when de-installing or moving a voice model from the speech recognition program to a storage medium. The user selects the voice model to be used and the destination where the voice model will be stored. When the user selects the OK button, a software execution process begins copying the user's parameters, and other files that make up the voice model. The process creates a catalog of the voice model system environment (parameters) and packages the files into a central location with a log file, information file, and initialization file to support future movement of the specific voice model. For VMM, more specifically, the sequence of events includes:

1. User clicks "Move voice model to Media" button.
2. A folder labeled "Users" is created on the destination media.
3. A "users.ini" file is created in the Users folder. This file is a logical translation from Username to a user file name that Dragon will open.
4. VMM then creates and writes the user specific registry information into a file called VMMinfo.txt
5. A "user" specific folder will be created in the Users folder. There are several files in the user specific folder.
6. The user as a result of the Dragon training process creates the files listed below. These files are copied to the CDWriter using the standard Dragon directory structure. Files included are:

audioin.dat

Current Folder containing:

topics (configuration file)
options (configuration file)

```
global.DVC
Voice folder
        DD10User.sig
        DD10User.usr
GeneralE Folder
        dd10voc1.voc
        dd10voc2.voc
        dd10voc3.voc
        dd10voc4.voc
        General.voc
Shared Folder
        archive.voc
```

[0032] These files and related registry parameters information make up the voice model for this example for the Dragon Professional application.

[0033] Figure 3 is the screen displayed when installing a voice model into voice recognition system. From this screen and the follow-up screen shown in Figure 4 the user selects a specific voice model and the location of the voice model to be used. When the user selects the OK button on Figure 4 a software execution process starts reading and executing the VMM initialization files and deposits the voice recognition parameters and files into the operating systems registry and parameter configurations files. The sequence of events occurs as follows:

1. VMM pops up a user selection window asking for drive containing the voice model.
2. After the drive is selected, VMM looks on the selected drive for the Users folder containing the voice models, specifically, the users.ini files.
3. If VMM does not find any users, a window pops up saying that no users were found with an OK button to click returning the user to the previous screen as illustrated in Figure 5.
4. If users exist, VMM then asks the user to select one of the voice models it found on the selected drive.

5. VMM then reads the file VMMInfo.txt file to determine the appropriate registry settings.
6. If the user already exists, VMM will prompt the user to ask if the existing user files should be overwritten.
7. If the user responds by clicking the OK button, then steps 8 onward will be executed, other wise VMM will go back to the main VMM screen.
8. If there is no user specific folder, then VMM creates the user specific folder in the standard Dragon hierarchical directory structure otherwise it uses the existing folder.
9. VMM then copies all the specific user files listed previously to the specific User folder.
10. VMM then configures the registry parameters for the selected user.

[0034] This methodology of voice model mobility could enhance many situations where training a voice recognition system is necessary, for example voice model mobility could enhance US Patent 6,477,491 Chandler et al. (System and Method for Providing Speaker Specific Records of Statements of Speakers) and Epstein et al. US patent number 6,327,343 (System and Method for Automatic Call and Data Transfer Processing) both mentioning the need of obtaining or assuming that a speaker voice model exists. Voice Model Mobility version 1.0 (VMM V1.0) is the first step toward the concept of modular plug-able voice models. This concept enables new features to be incorporated into voice models to provide enhancements on a wide variety of applications (e.g. security, learning aid for people speaking, singing, and language translation, voice mail to text, games, and guidance, etc.). While the concept of Voice Model Mobility could be applied with any voice recognition software, application or hardware, the VMM software application developed and used and described here is for example purposes and used Dragon Systems Voice Recognition Professional Application. The transfer medium could have been floppy disk, network, credit card strip, or other means of storing data.

[0035] The VMM prototype has been tested, debugged, and upgraded and is presently being used by many people. The current version works with high capacity floppy disk, network drives, CD media, and internet network drives, and other removable storage media. Given future data compression and larger media capacities, the goal would be to put Voice

Models on credit card type magnetic strips requiring personalized identification to enable the models similar to credit cards and bank ATM cards of today.

[0036] It is also the object of this invention to describe and provide a method for the translation of a Voice Model created in one voice recognition application (i.e. Dragon Voice Recognition Software) to be transferred and recognized by other LVVR applications (e.g. IBM ViaVoice, L&H, and Philips Speech Pro). The mechanics of translating voice models between LVVR applications include; 1) An information file is created identifying which parameters are needed for each type of LVVR systems. 2) The parameters are read from one LVVR system and translated to an LVVR common file format (.lvr). 3) The parameters are then formatted from the (.lvr) to the target voice recognition application format. 4) The file is then translated to the desired voce model format to create the final voice model. 5) The voice model is plugged-in to the destination LVVR system using the VMM techniques.

[0037] It is also the object of this invention to appreciate that services for storing, translating, and transferring voice models could be implement and a fee charged for such services. These services and/or software applications allow a user to have access to a proficient personalized Voice Model regardless of physical location or appliance being used. The voice model can be transferred electronically over wire or wireless transmission.

[0038] In yet another aspect of moving voice models between systems to enable mobility and a reduced system dependency, an alternate view is the ability to train a voice recognition system to recognized many uses with a single voice model, making the voice recognition system Speaker Independent. As stated earlier, when using Large Vocabulary Voice Recognition (LVVR) applications two problems are experienced. These are machine dependency and speaker dependency. Voice model mobility was a method to remove machine dependency. If many people want to use a specific voice recognition system for LVVR dictation using the current technology each person would have to train each machine to be used separately which is usually not feasible for the masses of people and potential numbers of systems.

[00039] It is the object of this invention to describe and provide a method for a new type of Voice Model called a Super Voice Model or SVM that has the ability to achieve speaker independent voice recognition. The technology that enables this ability is VMM for the movement of voice models combined with RAP Meter technology (described at a later

point in this document) to verify success and adjust parameters based on real time feedback. The key difference between voice models presently used and a Super Voice Model is: Current voice models attempt to adapt to the many different speakers, while a Super Voice Model attempts to adapt the speaker to fit a standard voice model (the Super Voice Model).

[0040] Referring to Figure 17, given that Voice Models will be available for transfer using VMM, they can be collected into a Voice Model Library (Figure 17 #104). By having a collection of voice models available for analysis and classification, a new type of synthetic voice model can be created (Figure 17 #107), derived from the parameters available in the collection (Figure 17 #105). In general, the Super Voice Model involves a quick sample of voice input from a new user (Figure 17 #106) and a comparison of the available parameters in the parameter lookup table (Figure 17 #105) and a selection of the specific parameters from the voice recognition voice model files (Figure 17 # 101, 102, 103) with the final output result a synthetic Voice Model (Figure 17 #107) to recognize the person speaking at the time without that person previously training the LVVR system. The new Voice Model can optionally be calculated real time or prior for a given person via some quick recording. The SVM is based upon having information about Voice Models readily available, organized, and ready to statistically calculate to create a synthesize Voice Model that can be used for any given speaker at the time. The overview process flow is as follow; VMM creates and archives voice models into a Voice Model Library (Figure 17 #104), the voice models are collected and categorized by common attributes of speakers, analysis' is implemented with results deposited into a table (Figure 17 #105) indicating availability of parameters for a potential synthetic generic voice model. As more voice models are added, potentially new generic voice models can be created. A Super Voice Model includes the library of voice models (Figure 17 #104), the lookup table of voice models indexed by parameters (Figure 17 #105), the logic to select a specific voice model from the table (Figure 17 # 108), and the ability to install the synthetic voice model into a voice recognition dictation system using VMM technology and a measurement of the success of the synthetic voice model using RAP Meter technology. A possible enhancement to the Super Voice Model database selection rules could include algorithms similar or as described in US patent 6,029,195 Herz (System for Customized Electronic Identification of Desirable Objects).

[0041] In summary, the "Super Voice Model" eliminates the need for specific user training or enrolling into a voice recognition system. The Super Voice Model is a classification and cataloging of many user voice models based on a library of known unique user files (Speaker Voice Models) classified by characteristics of each unique speaker including gender, rate of speech, and phoneme characteristics (e.g. spectral characteristics, pitch, frequency, and amplitude) and the use or combining of these files to statistically create a speaker voice model that previously did not exist or possibly the ability to use a speaker voice model that is very similar if one does exist.

[0042] As we look toward the future, computer voice recognition is moving toward noise recognition using one of many methods (e.g. training, pre-programming, learned through the experience of artificial intelligence or expert systems). If this technology is applied to the analysis of plane crashes, as an example, it could help to an understanding failures leading back to root causes. For example, building voice models of mechanical failure from previous audio recordings of mechanical malfunctions, during real time flights, sounds can be analyzed and warnings issues based on noises outside of the typical flight background noise.

[0043] It is yet another object of this invention to describe a methodology for measuring voice recognition accuracy and performance in combining them into a single metric to provide a reference point indicating voice recognition system reliability. Using multiple systems and transferring voice models can result in a degradation of system performance and text output accuracy. As a result indicators are needed to accurately measure accuracy as a correct word input count and performance as system responsiveness. These two key measurements performing reliably are what people expect from a quality voice recognition system. These metrics are defined here as Reliable Accuracy Performance Rate or "RAP Rate". When using multiple voice recognition systems, the expectation is to achieve comparison RAP Rates between systems. To accomplish a consistent RAP Rate an accurate measure must first be referenced. Then, based on the RAP metrics, decisions can be made as to the adequacy. If improvement is needed then, logical decisions can be made as to what changes affect RAP Rate.

[0044] Reliable Accuracy Performance Rate (RAP Rate) is defined in this invention as spoken audio correctly translated to text delivered with measured delay time from word spoken to word visible text in an electronic document. Components that affect RAP Rate include hardware components, software components, and a person or "user". From this, the following can be stated: Reliable Accuracy Performance Rate = User (u) + System (s) + Quality of components (q) + Integration (I) or RAP Rate = u + s + q + I. Where, a user "u" is defined as the person speaking to a voice recognition system. The system "s" is defined as a system trained to recognize a person's voice for the purposes of identifying audible spoken words. Quality of components "q" is defined as the hardware and software component functionality that is appropriate for voice recognition dictation applications and Integration as "I" defined as how the components are combined together including the merging of hardware, software, and parameters focusing on providing optimal voice recognition. For example, if a system has a reliable accuracy of 96% and a reliable performance of 1 second, then the RAP Rate would equal 96% at 1second or a RAP Rate of 96/1. Presently, a large vocabulary voice recognition dictation system including quality components and good integration can deliver a RAP rate of approximately 96% at 4 seconds (96/4). The RAP Rate equation components can be further defined:

[0045] Quality "q" Defined as compatibility of components and functionality that are well-matched for LVVR = *CPU margin % / (% of app in memory / (% of app in memory-(KB Cache / cache hit rate)/60) - (A/D conversion time + bus throughput latency).* As an example, this equation indicates aspects of hardware that can be changed to achieve an improved RAP rate focusing on the metric of *Performance* indicating system responsiveness to the process of voice recognition. The performance result is measured in time (seconds for current technology). The "delay" in the performance definition will never be zero. It may not be immediately perceivable to a user; but it can be measured and over periods of time will be perceived by the user.

[0046] Integration "I", defined as combining the components into a system environment to be used for voice recognition dictation where; *I = System parameters + Application parameters incompatibilities – Other task executed – Throughput resistance.* The integration aspect of RAP Rate affects reliable accuracy. System parameters include hardware (microphones, sound port, AD conversion devices, DSP methods, etc. and the related software driver modules, firmware, bios, operating systems, applications/utilities and their specific parameters). Computer parameters designed to accomplish general computing (i.e. word processing, multimedia, games, etc.) can contain parameters that are in conflict with a specific goal such as LVVR. Setting up software parameters to ensure the capabilities for LVVR are enabled at all levels can improve RAP Rate. As a result, "Integration" ties directly to RAP Rate.

[0047] To measure the specific metrics of accuracy and performance a RAP Meter software application is created. A person skilled in the art of computer hardware and software development will realize that other methods of creating a device to measure accuracy and performance for voice recognition are possible including a separate device using a separate hardware and software. The RAP Meter is one example used here to further describe the concept. Referencing Figures 6 through 10, an explanation of the RAP meter software is as follows:

[0048] Figure 6 represents the opening screen of the RAP Meter software. This screen contains 6 user control buttons and quick instructions in a display area of how to use the RAP Meter. The voice recognition software is launched and running in the background. The 6 control buttons include: 1) Verifying RAP, 2) Log filed, 3) Help button, 4) Selected mic, 5) Certificate, and 6) Exit button.

[0049] To operate the RAP Meter a user selects the verify RAP button. A new session screen is displayed as seen in Figure 7. The user inputs specific information about the test that is about to occur including a name for the test session, what kind of microphone is being used, and what is the voice recognition application software. The use also selects a sound port from the available sound ports displayed. After entering this information and clicking the "OK" button, another screen to implement the actual test will be displayed as shown in Figure 8. Referring to Figure 8 there are two separate areas of the screen that contain controls. The top area of the display operates the performance test in the low area of the display operates the accuracy test.

[0050] To perform a new performance test the user clicks the top new test button. This will cause words to be automatically displayed in the performance display area. As the words are displayed the user is to speak the words as they appear. The time is measured from the time the word is spoken until the point in time when the word would be displayed on a computer screen. This measured delay time is displayed for each word in the delay window. When the performance test is completed the word delays are averaged and written into a log file or wrap or a certificate. For performance, the RAP Meter records the time that sound was input (Tstart) and subtract from the time that text is displayed (Tend) on the screen for editing in a text application. Thus, Performance = Tend - Tstart.

[0051] To perform a new accuracy test the user clicks the new test button on the lower half of the RAP Meter display. In the text display area a paragraph is printed out for the user to read. As the user reads the text, the RAP Meter counts the words that are translated correctly versus the words that are translated incorrectly. It calculates the ratio of these two values in displays the results as a percentage of correct words in a log file and RAP Rate certificate. As a person reads the text the words are translated into text. The RAP Meter compares the original text with the text that was translated using the LVVR system and responds back to the user with an accuracy measurement in percentage of correct translated words (e.g. 96 % correct). Thus, Accuracy % = words incorrect / words correct. Incorrect words can be highlighted for display to the user.

[0052] To view and use the results of the performance and accuracy test the user can select the log file button. A log file will be display as shown in Figure 10 indicating the results of the performance in the accuracy test in combining the metrics to create a RAP Rate of accuracy percentage versus performance as system response time. The log file includes specific details on system specifics including voice recognition application, computer hardware (processor type, memory capacity, CPU speed, etc.). The log also includes words that were spoken during the performance test and an indication if the words were correctly translated along with the delay for each word spoken measured in seconds or fractions thereof. The log file also includes output of the accuracy test including the paragraph that was read as displayed on the screen and the paragraph that was spoken by the user of the test (input and output text). The accuracy metric is also included in the log file as a percentage of correct words versus and correct words. The log file can be printed, sent to the mail, or transferred by other means.

[0053] Referring back to Figure 6, there is a "Certificate" button. Clicking this button will display a RAP Rate certificate of the current or previous sessions as shown in Figure 9. Again, referring back to Figure 6 if the user clicks the "Select mic" button, another screen is displayed showing the microphones available on the system to be used and a VU Meter for a quick check to ensure the microphone is working correctly. Clicking the help button displays a help file on RAP meter features and use.

[0054] The RAP Rate Meter concept is an extend feature for verification of the Voice Model Mobility concept providing an indication of successful operation of moving a voice model. The RAP meter can also be provided as a separate application for certification of LVVR applications. As an example, companies could use RAP Rate to advertise displaying RAP certification and could be charged a per-system usage fee or license fee as a way to create revenues from RAP licensing.

[0055] The components (Application Programmers Interfaces) used to get real time capture and performance measurement includes; software application handles to indicate applications loaded are used for LVVR. Audio input device IRQ and I/O address range and software driver IO function calls to indicate when the A/D translation has started. Speech recognition function calls (e.g. RealTimeGet and TimeOutGet) to indicate when the voice recognition engine has started and completed the translation. Video board IRQ and I/O address range and software driver IO function calls to determine when the text is being displayed to the editor on the screen. As words are spoken into a microphone, trigger points are set to indicate when each section of the text translation and display process has completed its task. The following steps indicate one method regarding how RAP meters can function as an example:

1. (Setup) Application used for LVVR is identified
2. (Tstart) A/D time is measured by logging the time the driver gets sound input. This can be accomplished through a peek message or for MSWindows ;InmChannelAudio::IsIncoming, HRESULT IsIncoming(void); or other method.
3. (Pstart) Determine and log the when the speech processing engine has received the sound by using a function call (i.e. RealTimeGet).
4. (Pend) Determine and log the time when the speech engine has completed the translation using a function call (i.e. TimeOutGet).

5. (Tend) Determine when the graphics driver has displayed the text using a peek message or for MSWindows a function call (i.e. UI Text Event; TEXT_VALUE_CHANGED).
6. (Report) Calculate the times. For general performance Tend - Tstart will supply the performance delay. For further resolution to determine areas of throughput resistance, steps 2 and 3 can be used.

[0056] This is one method of achieving the desired goal of measuring RAP Rate; however one skilled in the art of software programming or similar computer skills sets can realize other methods of achieving the same results. This method used as an example is not intended to be the only possible method and it is recognized that many alternative methods are possible.

[0057] In accordance with the features previously described in this invention the RAP meter can be applied to engineering and manufacturing processes. It is the object of this invention to describe a methodology for a process that measures specific hardware and software features necessary to support optimal Large Vocabulary Voice Recognition (LVVR) using Reliable Accuracy Performance (RAP) Rate as the measurement reference of the processes. during engineering the reference measurement values are inserted into process sheets allowing controlled steps to be followed. Using this technique processes can be developed for a production line for LVVR system manufacture. For development the methods include a hardware components selection process based on investigation of functions needed, a test process to measure components adequacy, and documenting functionality and parameters.

[0058] Referring to Figure 14 for reference, the engineering development process is described. Figure 14, #501 starts the process with a component investigation of optimal components to be used in voice recognition dictation systems. Supplier components are investigated for the specific hardware functionality needed (Figure 14, #502). Specifications and documentation distributed by component suppliers is investigated for the specific fit to the RAP list of requirements Figure 14, #506. The process can include having the suppliers of hardware components produce the list of hardware that meets the requirements of the RAP list. At present the following is a list as an example of optimal features for voice recognition dictation systems:

Optimal features to enhance RAP Rate

High-speed microprocessors

Robust floating point features

Large on chip and off chip cache memory 512 kb or more

High-capacity/fast main memory (optimal 512 megabytes)

Sound input device with performance focused on input in the range of the human voice

An operating system specifically configured (tuned) for the application of voice recognition including:

Removing any throughput resistance including processes that require main CPU clock cycles but don't provide advantage to LVVR.

Removing operating system resources that use main memory or run in the background like schedulers, virus checking, or utilities that execute polling at specific time intervals or triggers.

Removing applications that use main CPU floating point and moving that work to other microprocessors.

Ensuring that any operating system or applications being used return allocated memory back to being available and not left locked out by the LVVR application.

[0059] These specific features are typically not included recommendations with the "off the shelf" hardware and software. Voice recognition application packages from vendors like IBM, Dragon, or L&H will not mentioned all the details as part of the software requirements. These features are dedicated to the task of voice recognition and can be packaged as such to create a large vocabulary voice recognition appliance. Originally, to measure and characterize the hardware for large vocabulary voice recognition experimentation, tools and indicators that were readily available in the industry and part of the operating systems were used. Using these tools measurements could be acquired and a determination was made as to hardware resources needed. Then a manual process of measurement was used in effort to refine what further hardware and software parameters would be best. From this work an automated test methodology was built to allow production mode for development and manufacturing to be put in place to characterize the hardware faster. The automated test methodology is using the RAP Meter technology. Figure 14, #503 illustrates the RAP Meter test and verification being applied to the engineering development process. If an acceptable RAP Rate is achieved the documented process sheets will be delivered to the manufacturing process as illustrated in Figure 14, #508, otherwise adjustments to components and/or parameters are made Figure 14, #505 in the system goes back to test Figure 14, #503. An example engineering process sheet for system BIOS is shown in Figure 16.

[0060] Referencing Figure 15, a description of the manufacturing processes provided. Manufacturing process sheets (Figure 15, # 607) are provided from the engineering process in Figure 14. The components are ordered (Figure 15, #601) as described by the engineering process sheets. The components are integrated into a system package (Figure 15, #602) and then tested using the RAP Rate technology (Figure 15, #603). If the RAP Meter test indicates successful results, the system is packaged for shipping (Figure 15, #606) otherwise the processes updated (Figure 15, #605) to avoid unsuccessful results in the future.

[0061] These processes were developed around the concept of achieving a standardized level of Reliable Accuracy and Performance Rate (RAP Rate). The components and functionality needed to achieving an adequate RAP Rate are listed and explained below as an example and expected to change over time. However, as the components and technology continue to improve, the standard measurement of RAP rate will remain a valid measurement allowing users to understand what is being purchased or provided. In other words RAP Rate is to voice recognition what wattage or power measurement is to the electric industry.

[0062] Description of the features in additional detail: A RAP rate of 100/0 defined as 100 % accuracy with zero delay time measure from time of spoken word to displayed text in an application is the ultimate goal. It was determined from research and testing that specific components can affect RAP rate. Additionally where some component may be lacking, another may be more than adequate resulting in a similar RAP rate. As described earlier as part of this invention, RAP Rate = u + s + q + I. The following list explains items that are important with regards to these metrics and list some industry standard methods of measurement. When RAP rate is not acceptable the other methods can be used to isolate the problem areas. High-speed microprocessor; presently Microprocessor speeds today are up to 800 MHz+ on average and steadily moving to higher processor speeds. When measuring microprocessor usage while using LVVR applications, results show that microprocessor usage is at 100%. To determine this, a combination of a manual process and an automated process is used. One method of measuring CPU usage is by using the performance monitor tools available with an operating system like Microsoft Windows 98. The goal is to achieve a margin of microprocessor resources left available while dictation to a system is being done. Ideally, with voice recognition a performance in the range of no noticeable delay from the time the words are spoken to the time the text is displayed in a text editor is a desired metric. If other applications are to be run simultaneously, then an additional margin in performance must be added to avoid affecting RAP rate.

[0063] Robust floating point features; a robust floating-point microprocessor is needed due to the intensity of math calculations that are routine for voice recognition applications. Floating point microprocessors may be embedded in a main microprocessor or done separately by the main CPU instruction set or software. Microprocessors that support floating-point in different ways can directly affect RAP rate. Ideally a microprocessor that has a combination of hardware registers, floating point instruction set with features that allow multiple calculations with minimal clock cycles, while supporting access to fast cache memory are desirable. Measurements on floating points can be achieved using industry standard tools or published results in the trade magazines or from the manufacturers.

[0064] Large on chip and off chip cache memory; Cache memory is the closest storage medium to the microprocessors doing the work. Typically the memory closest to the main CPU will be the fastest data access. The capacity of the cache memory, the percentage of cache hits, and if the cache is embedded in the CPU chip or off chip will make a difference. "*KB Cache / cache hit rate*" work as performance enhancement in the equation and can be measured using embedded OS performance tools of Microsoft Windows.

[0065] High-capacity/fast main memory; a large capacity main memory is desired and will affect performance. Enough capacity to allow the LVVR and related applications to execute directly out of memory yields the best performance. Having to go out to disk is a magnitude of time longer and should be avoided whenever possible. Testing and measuring results indicate that using a LVVR system can easily use 256 megabytes to prevent disk access. This can be measured using operating system tools like the performance monitor of Microsoft Windows 98, along with other tools available in the computer industry. As memory is reduced a delay resulting in a lower RAP rate will occur. Therefore the equation includes a metric "*% of application in memory*" as add or minus to performance. These values will change over time and technology, but the goal remains the same for LVVR, to execute without disk access.

[0066] Sound input device with performance focused in the range of the human voice. Most sound components for PC's focus on output while input is a secondary consideration. Therefore sound input components can cause performance problems. The physical system interface/bus can also add or subtract to performance. *A/D conversion time + bus throughput latency* time subtracts from the performance and can never be removed from the system. While this delay can be lowered to the level of not perceivable, it will never be reduced to zero. Oscilloscopes are one method of measuring this delay. This measurement is also included in the performance measurement of RAP Rate which can be measured through a software tool like a RAP meter.

[0067] When the objective of Quality is completed, then integration of the component parameters and reduction in bottle necks are the objective. The integration aspect of RAP rate can be affected by software (firmware, operating systems, applications/utilities and parameters). Parameters can enhance or subtract RAP rate from a large vocabulary voice recognition application. As an example, a word processing application with a parameter set to auto correct grammar during dictation may cause sever RAP rate reduction due to resources being shared for real time grammar correction and LVVR. Starting at the lowest level (BIOS) and working through the OS towards the LVVR application is one method of tuning software for a good RAP rate. Another method would be to reverse the order and start at the LVVR application and work back. Then create a software utility that does the parameter settings automatically based on the known information. Therefore an explanation of the equation above would be to add items that can be modified to enhance LVVR and to subtract items that cannot be removed and must be worked around like incompatibilities. There are not industry standard tools to measure specifically for types of parameters that affect RAP Rate. At this point RAP Rate or an individual component of RAP is the only measurement that sums these conclusions specifically voice recognition dictation systems. .

[0068] It is the object of this invention to describe a methodology for developing a handheld transcriber with full features to execute voice recognition software like Dragon naturally speaking internal to the hand held transcriber. Using the RAP Rate engineering process, a handheld transcriber with voice recognition software internal to the transcriber was developed. In accordance with this invention and the uses for the RAP Meter and Voice Model Mobility technology, it is the object to describe a methodology to an improvement of the present handheld transcriber technology used for voice recognition dictation.

[0069] One object of this invention is to describe a method that can reduced voice recognition system training time by using a cable that allows a microphone to be connected to both the handheld transcriber and a desktop PC and implementing a process of synchronization for training large vocabulary voice recognition on both devices simultaneously. Referring to Figure 11, a Y cable configuration connects to the handheld transcriber microphone input and at the same time connects to the computer for the voice recognition software input. This configuration creates a single microphone input for both devices (the computer and the handheld transcriber). Using this method enables a single training session for both devices. This was successful allowing a user to train a hand held transcriber at the same time the desktop system was trained. This method saved the redundant training time but did not solve the other problems of display for direct user feedback and the ability to correct accuracy errors during the dictation process. This training once for two devices has advantages over the method described in US Patent 6,477,493 Brooks Et Al. (Offsite Voice Enrollment on a Transcription Device for Speech Recognition) where it is described how to train a handheld transcription device using a cable connected to a voice recognition system. Using Brooks et al. system requires training once for the transcription device and a second time for the desktop voice recognition system.

[0070] A better method of accomplishing large vocabulary voice recognition for handheld transcribers would be to package the desktop system hardware into a handheld form factor.

[0071] It is further the object of this invention to describe a methodology of building a handheld device containing enough CPU power, memory, storage, display, input sound system, and supporting operating system and voice recognition software Inc. into the handheld device to accomplish voice recognition dictation with the handheld device.

[0072] From previous work it was discovered that computer hardware to support fully functional large vocabulary voice recognition for handheld transcribers and these types of applications must include at least the following components to be effective: High-speed microprocessor with robust floating point feature, large on chip and off chip cache, high-capacity/fast main memory, quality sound input device with performance focused in the range of the human voice or signal to be translated, and an operating system specifically configured (tuned) for the application of voice recognition. Components were found, combined, tested, and packaged in a small form factor. These smaller components allowed a prototype to be built in 1999 to prove the viability of such a device. Referring to Figure 12 is a graphic a prototype large vocabulary dictation handheld transcriber using Microsoft Windows 98 and Dragon Professional Voice Dictation application as an example. Other operating systems and applications could have been used like Linux and a public domain voice recognition application. This device solved the problems listed above. The prototype is a fully functioning handheld transcriber focusing on proof of the concepts of form factor, use of VMM via a network drive, the ability to provide direct feedback of speech to text while dictating in the handheld environment, and the ability to use a voice recognition interface combined with touch screen for user control. The prototype supports a vocabulary of over 30,000 words. Test results from this proto-type indicate that production models could support large vocabularies including libraries to support medical and legal services. This prototype includes battery operation, network connection, USB, keyboard and mouse if desired, and connection for 120 volt AC power, and a microphone input jack.

[0073] Referring to Figure 13 is a block diagram of the handheld transcriber components. They include a 400MHz CPU (Figure 12, #200), 256KB cache (Figure 12, #201), 256MB of memory (Figure 12, #203), and Neo Magic graphic chip (Figure 12, #204), a PCI to ISA bridge (Figure 12, #205), and a USB port (Figure 12, #209). These components are integrated into any Plug n Run motherboard/daughter board configuration purchased from Cell Computing, located in California. The A/D converter (Figure 12, #208) was purchased from Telex in the microphone (Figure 12, #207) was purchased from Radio Shack. The color LCD (Figure 12, #206) was purchased from Sharp.

[0074] The Microsoft Windows 98 operating system was loaded onto the IDE disk drive in the voice recognition software (Dragon Naturally Speaking) was installed onto the handheld transcriber.

[0075] After applying power to the device it can be controlled using voice recognition commands and touch screen. When the device becomes ready it automatically is in a mode to select a user and dictation can start. Dictating to machine; the device supports a microphone input jack with a microphone on/off switch that can be momentary or left in either mode. The user speaks into a microphone and the voice is translated into a text editor on a handheld screen. What makes this handheld device unique is the amount of words (large vocabulary of greater than 30,000 + words) that can be translated realtime during dictation. Save file saves the dictated text files for later editing, importing and exporting, archival or transfer. The device supports a network connection for moving the files and voice models to and from the handheld device.

[0076] When Windows 98 boots the startup configuration launches Dragon voice recognition software with a Dragon pad work process and turns the microphone on. The user is then able to use voice commands to navigate the Windows operating system and the Dragon software and also dictate to the Dragon pad work processor. The RAP Meter technology was used for selection in verification of components to achieve the desired goal of handheld voice recognition dictation. Voice model mobility was used to maintain an up-to-date voice model between the handheld device in the desktop voice recognition system. This is one example of a handheld transcriber using components to support the voice recognition software internal to the handheld transcriber. It is to be appreciated that other components could be used to achieve the same results.